Com^{2}MaC Workshop on Cryptography

26-28 june 2000 - Pohang - South Korea

Secure Designs for Public-Key Cryptography based on the Discrete Logarithm

David Pointcheval
Département d 'Informatique
ENS - CNRS

David.Pointcheval@ens.fr http://www.di.ens.fr/~pointche

Overview

- Introduction
- Security Arguments
- Signature
- Encryption
- Conclusion

Com^{2}MaC Workshop on Cryptography

26-28 june 2000 - Pohang - South Korea

Introduction

David Pointcheval
Département d 'Informatique
ENS - CNRS

David.Pointcheval@ens.fr   http://www.di.ens.fr/~pointche

Cryptography

Cryptography:
to solve security concerns

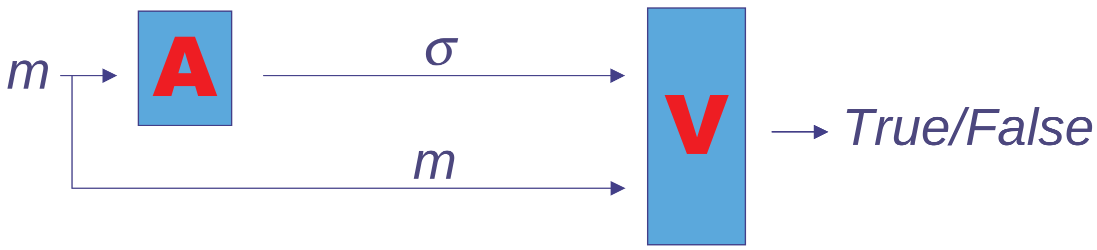
Authentication
Integrity
} ⇒ signature

Confidentiality ⇒ encryption

Authentication/Integrity

Authentication Algorithm **A**

Verification Algorithm **V**

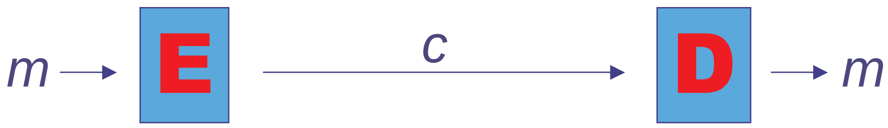


Security: it is impossible to produce a new valid pair (m,σ)

Encryption

Encryption Algorithm **E**

Decryption Algorithm **D**



Security: it is impossible to get back m just from c

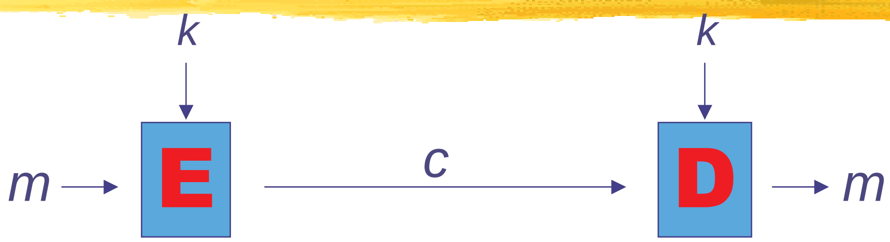
Foundations

To build such primitives, one needs *(trapdoor)* ***one-way functions***:

$x \rightarrow y = f(x)$ is easy (Encryption, Verification)

$y = f(x) \rightarrow x$ is difficult (Decryption, Signature)

Conventional Cryptography



f is an intricate network of permutations/substitutions, parameterized by a secret key

$\mathbf{E}_k = f_k$

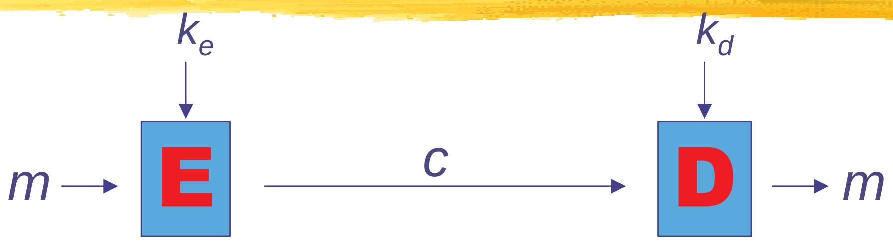
$\mathbf{D}_k = f_k^{-1}$

f_k and f_k^{-1} are both "easy" to compute with k

f_k and f_k^{-1} are both "difficult" to compute without k

difficult: heuristic!

Modern Cryptography



f is a non P-problem (no polynomial algorithm)

$\mathbf{E}_{k_e}(x)$ = instance I of f from k_e,
for which x is a solution

$\mathbf{D}_{k_d}(I)$ = solution of I

"easy" to build an instance with a known solution
"difficult" to solve an instance (but easy with k_d)

difficult: complexity theory

One-Way Functions

- **NP**-complete problems:
  - hard in the worst-case
    what about the average case?
  - hard asymptotically
    what about the difficulty of instances of reasonable size (few bytes)?

  $\Rightarrow$ quite few candidates (for signature)
- Number Theory:
  - factorization $\Rightarrow$ RSA, etc
  - discrete logarithm $\Rightarrow$ Diffie-Hellman, etc

The Discrete Logarithm

- Let **G** = (<g>, $\times$) be any cyclic group of order q (noted multiplicatively)
- For any $y \in$ **G**, one defines

$$\mathrm{Log}_g(y) = \min\{x > 0 \mid y = g^x\}$$

- *One-way function*
  - $x \rightarrow y = g^x$ easy
  - $y = g^x \rightarrow x$ seems difficult

Various Groups

G = sub-group of

- $\mathbf{Z}_p^*$, $\mathbf{Z}_n^*$
  $\Rightarrow$ sub-exponential (NFS)
- an elliptic curve
  $\Rightarrow$ exponential (in general)
- a Jacobian
  $\Rightarrow$ exponential (in general)
- other
  - ideals of number fields (NICE)
  - braid group, …

Any Trapdoor …?

- The Discrete Logarithm is difficult

But no information could make it easier!

- The Diffie-Hellman Problem (1976):
  - Given $A=g^a$ and $B=g^b$
  - Compute DH(A,B) $= C=g^{ab}$

Clearly DH ≤ DL: with $a=\mathrm{Log}_g A$, $C=B^a$

C-DH Assumption:
the DH-problem is intractable

Another DL-based Problem

The **Decisional Diffie-Hellman Problem**:

- Given A, B and C in $<g>$
- Decide whether C = DH(A,B)

Clearly D-DH ≤ DH ≤ DL

D-DH Assumption:
the D-DH-problem is intractable

Application: El Gamal Encryption

- **G** = (<g>, $\times$) group of order q
- x : **secret** key
- $y=g^x$: **public** key

public	$\mathbf{E}(m) = (g^a, y^a m) \rightarrow (c, d)$

secret	$\mathbf{D}(c, d) = d / c^x$

One-Wayness = C-DH
Semantic Security = D-DH

Com^{2}MaC Workshop on Cryptography

26-28 june 2000 - Pohang - South Korea

Security Arguments

David Pointcheval
Département d 'Informatique
ENS - CNRS

David.Pointcheval@ens.fr   http://www.di.ens.fr/~pointche

Security Notions

Depending on the security concerns, one defines

- the goals that an adversary may would like to reach
- the means/information available for the adversary

Security Proofs

One provides a reduction from a “difficult” problem *P* to an attack *Atk*:

- $\mathcal{A}$ reaches the “prohibited” goals
  $\Rightarrow$ $\mathcal{A}$ can be used to break *P*
- no further hypothesis: standard model
- but that rarely leads to efficiency!
  $\Rightarrow$ some assumptions

Security Arguments

One provides a reduction from a “difficult” problem *P* to an attack *Atk*, under some ideal assumptions:

- ideal random hash function:
  random oracle model
- ideal symmetric encryption:
  ideal cipher model
- ideal group:
  generic model (generic adversaries)

The weakest: Random Oracle Model (ROM)

Com^{2}MaC Workshop on Cryptography

26-28 june 2000 - Pohang - South Korea

Signature

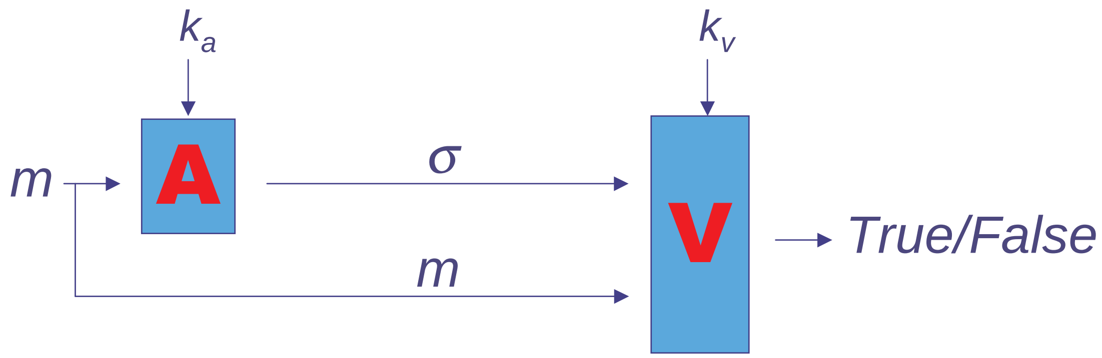
David Pointcheval
Département d ’Informatique
ENS - CNRS

David.Pointcheval@ens.fr http://www.di.ens.fr/~pointche

Authentication

Authentication Algorithm **A**

Verification Algorithm **V**



Security: it is impossible to produce a new valid pair (m,σ)

Security Notions

Total Break:
to recover the secret key

Universal Forgery:
to be able to sign any message

Existential Forgery:
to produce a new valid pair (m,σ)
(possibly m is without any meaning)

Kinds of Attacks

no-message:
  the adversary just knows the public key

known-message:
  she knows some message-signature pairs

(adaptively) chosen-message
  she has access to a signature oracle

Secure Signature

A Signature Scheme is said SECURE
  if it prevents
    any existential forgery
  even under
    adaptively chosen-message attacks

Then, the signature guarantees:

- the identity of the sender
- the non-repudiation:
  the sender won't be able to deny it later

Schnorr's Signature (1989)

G $= \langle g \rangle$, q and g : **common data**
x : **secret** key $y=g^x$: **public** key

Signature of the message m :
choose a random $k \in \mathbf{Z}_q$
compute $r=g^k$
get $e=h(m,r)$ and $s = k\text{-}xe \bmod q$

$\sigma = (e,s)$

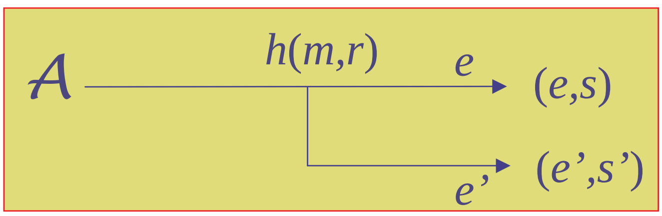
Verification of (m,σ) : $u = g^s y^e \; (= g^{k-xe} g^{xe})$
test whether $e=h(m,u)$?

Security?

Existential Forgery
under chosen-message attacks
(in the random oracle model)
= computation of discrete logarithms

(Pointcheval-Stern EC '96)

Idea: *Forking Lemma*



$$g^s y^e = r = g^{s'} y^{e'}$$
$$\Rightarrow g^{s-s'} = y^{e'-e}$$

Let $\alpha = (s\text{-}s')/(e'\text{-}e) \bmod q \Rightarrow y=g^\alpha$

Trusted El Gamal Type Signatures Schemes (BPVY PKC '00)

Key-Gen: $x \in \mathbb{Z}_q$ and $y = g^x$

- Two hash functions G and H
- F_1: $\mathbb{Z}_q \times \mathbb{Z}_q \times \mathbf{G} \times \mathbf{H} \to \mathbb{Z}_q$
- F_2: $\mathbb{Z}_q \times \mathbf{G} \times \mathbf{H} \to \mathbb{Z}_q$
- F_3: $\mathbb{Z}_q \times \mathbf{G} \times \mathbf{H} \to \mathbb{Z}_q$

such that, for all $(k,x,t,u) \in \mathbb{Z}_q \times \mathbb{Z}_q \times \mathbf{G} \times \mathbf{H}$

$$F_2(F_1(k,x,t,u),t,u) + x\, F_3(F_1(k,x,t,u),t,u) = k \bmod q$$

$\Rightarrow g^{E_g} y^{E_y} = g^k$ where $s = F_1(k,x,t,u)$

$E_g = F_2(s,t,u)$ and $E_y = F_3(s,t,u)$

TEGTSS - I

Sign(m): $k \in \mathbb{Z}_q^*$ and $r = g^k$

$t = G(m)$ and $u = H(r)$

then $s = F_1(k,x,t,u)$ $\quad \to \sigma = (s,t,u)$

Ver(m,σ): check if $t = G(m)$ and $u = H(w)$,

where $w = g^{E_g} y^{E_y}$

with $E_g = F_2(s,t,u)$ and $E_y = F_3(s,t,u)$

and 2 further properties…

TEGTSS - I: Security

KCDSA: $F_1(k,x,t,u) = (k - t \oplus u)/x \bmod q$

$F_2(s,t,u) = t \oplus u \bmod q$

and $F_3(s,t,u) = s \bmod q$

Security Claim:

If H behaves like a random oracle

but G is just collision-resistant then

existential forgery = extraction of x

Proof:

use of the Forking Lemma [PS96]

TEGTSS - II

Sign(m): $k \in \mathbf{Z}_q^*$ and $r = g^k$

$t = G(r)$ and $u = H(m,t)$

then $s = F_1(k,x,t,u)$ $\rightarrow \sigma = (s,t,u)$

Ver(m,σ): check if $t = G(w)$ and $u = H(m,t)$,

where $w = g^{E_g} y^{E_y}$

with $E_g = F_2(s,t,u)$ and $E_y = F_3(s,t,u)$

and a further property

TEGTSS - II: Security

DSA-II: $F_1(k,x,t,u) = (u + xt)/k \bmod q$
$F_2(s,t,u) = u/s \bmod q$
and $F_3(s,t,u) = t/s \bmod q$

Security Claim:

If H behaves like a random oracle, but
- $x \to G(x)$ is $(l + 1)$-collision-resistant
- **OR** $x \to G(g^x)$ is $(l + 1)$-collision-free

then existential forgery = extraction of x

Proof: an improved forking lemma

Applications: KCDSA

KCDSA:

- provably secure
  if both G and H behave
  like random oracles

But one can weaken assumptions:

- provably secure
  if H behaves like a random oracle
  but G just collision-resistant

Com²MaC Workshop on Cryptography

26-28 june 2000 - Pohang - South Korea

Encryption

David Pointcheval
Département d 'Informatique
ENS - CNRS

David.Pointcheval@ens.fr http://www.di.ens.fr/~pointche

Asymmetric Encryption

Encryption Algorithm **E**

Decryption Algorithm **D**

$$\begin{tikzcd} & k_e \arrow[d] & & k_d \arrow[d] & \\ m \arrow[r] & \boxed{\mathbf{E}} \arrow[rr, "c"] & & \boxed{\mathbf{D}} \arrow[r] & m \end{tikzcd}$$

Security: it is impossible to get back m just from c, k_e, **E** and **D** (without k_d)

Better security?

- Perfect Security:

  the ciphertext and public data do not reveal any information about the plaintext (but maybe the size)

  Information Theoretical sense $\Rightarrow$ Impossible

- Semantic Security (Indistinguishability):

  no polynomial adversary can learn any information about the plaintext from the ciphertext and public data (but the size)

Kinds of Attacks

- Chosen Plaintext: *(basic scenario)*

  in the public-key setting, any adversary can get the encryption of any plaintext of her choice (by encrypting it by herself)

- Chosen Ciphertext (adaptively):

  the adversary has furthermore access to a decryption oracle which decrypts any ciphertext of her choice, but the specific challenge

Required Security

- OW-CPA: *(basic level of security)*
  - enough in some scenarios
  - not enough in many others
- CC-Attacks easy to perform

  $\Rightarrow$ attack to be made unuseful
- Plaintext-space often limited

  ("sell" - "buy" -- "yes" - "no" -- …)

  $\Rightarrow$ IND very often required

Main Security Notions

- OW-CPA: *(the weakest)*

$$\boxed{\Pr_{m,r}\left[\mathcal{A}(c) = m \,\middle|\, c = \mathbf{E}(m;r)\right]} \quad = \text{Succ negligible}$$

- IND-CCA: *(the strongest* - BDPR C '98*)*

$$\boxed{2\Pr_{r,b}\left[\mathcal{A}_2^{\mathcal{D}}(m_0, m_1, c, s) = b \,\middle|\, \begin{matrix}(m_0, m_1, s) \leftarrow \mathcal{A}_1^{\mathcal{D}}(k_e) \\ c \leftarrow \mathbf{E}(m_b, r)\end{matrix}\right] - 1}$$

= Adv negligible

DL-based Cryptosystems

- El Gamal:
  - OW-CPA = C-DH
  - IND-CPA = D-DH
  - CCA ? No because of malleability
- Cramer-Shoup:
  - IND-CCA = D-DH
- PSEC (Okamoto-Fujisaki-Morita):
  - PSEC-1: IND-CCA = D-DH (+ROM)
  - PSEC-2: IND-CCA = C-DH (+ROM)

Generic Conversions

- Any trapdoor one-way function leads to a OW-CPA cryptosystem
- But OW-CPA not enough
- How to reach IND-CCA ?

  $\Rightarrow$ generic conversions

  from OW-CPA to IND-CCA

Conversions (1/3)

- OAEP (Bellare-Rogaway EC '94)

  optimal conversion of
  any ***trapdoor one-way permutation***
  into an IND-CCA cryptosystem

  Efficiency: optimal (just 2 more hashings)

  Application: RSA
  (the sole candidate as
  trapdoor one-way permutation!)

Conversions (2/3)

- Fujisaki-Okamoto (PKC '99)

  conversion of
  any ***IND-CPA cryptosystem***
  into an IND-CCA cryptosystem

  Drawback: security relative to decisional
  problems (D-DH, Higher Residuosity, …)

  Efficiency:
  - optimal encryption (just 2 more hashings)
  - non-optimal decryption (1 re-encryption)

Conversions (3/3)

- Fujisaki-Okamoto (Crypto '99)
  Pointcheval (PKC '00)

  conversions of
  any ***OW-CPA cryptosystem***
  into an IND-CCA cryptosystem

  Advantage: security relative to computational problems (C-DH, Factorization, …)

  Efficiency:
  - optimal encryption (just 2 more hashings)
  - non-optimal decryption (1 re-encryption)

PSEC - OCAC

- PSEC 1: Fujisaki-Okamoto (PKC'99)
  conversion applied on El Gamal
  for which IND-CPA = D-DH
- PSEC 2: Fujisaki-Okamoto (Crypto'99)
  conversion applied on El Gamal
  for which OW-CPA = C-DH
- PSEC 3: Okamoto-Pointcheval
  new conversion (OCAC) which makes
  any OW-PCA cryptosystem
  into an IND-CCA cryptosystem

A New Attack: PCA

- **Plaintext Checking Attack:** the adversary
  - can get the encryption of any plaintext of her choice (by encrypting it by herself)
  - has furthermore access to an oracle which, on input a pair (m,c), answers whether c encrypts m, or not

Remark: IND-PCA cannot be achieved

$\Rightarrow$ we will just be interested in OW-PCA

A New DL-based Problem: G-DH

The Diffie-Hellman Problems:

- computational
  - Given $A=g^a$ and $B=g^b$
  - Compute DH$(A,B) = C=g^{ab}$
- decisional
  - Given A, B and C in $<g>$
  - Decide whether $C =$ DH(A,B)
- Gap
  - Solve the computational problem, with access to a decisional oracle

Intractability of the Gap-DH

The Computational Diffie-Hellman problem is believed intractable for suitable groups

Gap-DH easy $\Rightarrow$ D-DH = C-DH

D-DH easy $\Rightarrow$ G-DH = C-DH

The Computational Diffie-Hellman problem is believed strictly stronger than the Decisional version $\Rightarrow$ G-DH intractable

El Gamal OW-PCA = G-DH

PSEC - 3

- G and H: two hash functions
- E, D: symmetric encryption scheme

x : **secret** key

$y=g^x$: **public** key

$$\mathbf{E}(m): a \leftarrow_R \mathbb{Z}_q, R \leftarrow_R \mathbb{G}$$
$$A \leftarrow g^a, \quad A' \leftarrow R\, y^a$$
$$k \leftarrow G(R),\ B \leftarrow \mathsf{E}_k(m),$$
$$C \leftarrow H(A, A', R, m)$$

$\longrightarrow (A, A', B, C)$

$$\mathbf{D}(A,A',B,C): R \leftarrow A'/A^x,$$
$$k \leftarrow G(R),\ m \leftarrow \mathsf{D}_k(B),$$
$$\text{check whether } C = H(A, A', R, m)$$

Security Result

One just needs a symmetric encryption semantically secure against passive attacks:

- One-Time Pad: perfectly secure ($\mathsf{Adv}^E = 0$)
- Any classical scheme (DES, IDEA, AES,…)

  $\mathsf{Adv}^E = \nu$ (very small)

If an adversary $\mathcal{A}$ against IND-CCA reaches an advantage $\mathsf{Adv}^{\mathcal{A}} > \mathsf{Adv}^E$ one can break the Gap-DH problem with probability greater than

$(\mathsf{Adv}^{\mathcal{A}} - \mathsf{Adv}^E)/2 - q_D/2^{l_H}$

Semantic Security (OTP)

Given $A \leftarrow g^a$, $A' \leftarrow R\, y^a = R \,.\, \mathrm{DH}(A,y)$

$k \leftarrow G(R)$, $B \leftarrow k \oplus m$, $C \leftarrow H(A, A', R, m)$

In order to guess b such that $m = m_b$ an adversary has to ask either

- R to G to get k (and check B)
- (A,A',R,m) to H (and check C)

because of the randomness of G and H

Probability that R $(=A'/\mathrm{DH}(A,y))$ has been asked to G or H greater than $\mathsf{Adv}^{\mathcal{A}}/2$

Plaintext Extractor

Plaintext-Awareness (Bellare-Rogaway EC'94)

(A,A',B,C) ciphertext valid $\Rightarrow$ one has asked (A,A',R,m) to H to get a valid C

(but with probability less than $1/2^{l_H}$)

The plaintext extractor, to decrypt a given ciphertext (A,A',B,C), looks, for any query (A,A',R,m) to H which leads to C, whether

- $R = A'/\mathrm{DH}(A,y)$ (thanks to the DDH-oracle)
- $B = \mathsf{E}_k(m)$ for $k = G(R)$

Correct extraction with probability greater than $1 - 1/2^{l_H}$

CCA Security

After q_D queries to the decryption oracle

- all the decryptions are correctly simulated with probability greater than

$$(1 - 1/2^{l_H})^{q_D} \geq 1 - q_D / 2^{l_H}$$

- R has been asked to G or H with probability greater than

$$\mathrm{Adv}^{\mathcal{A}} - \frac{q_D}{2^{l_H}}$$

Properties of PSEC-3

- this is a new EG-scheme:
  - OW-CPA = C-DH (+ROM)
  - OW-PCA = Gap-DH (+ROM)
  - IND-CCA = Gap-DH (+ROM)
- hybridity: one can integrate
  any symmetric encryption scheme,
  semantically secure
  against passive attacks
  (a very weak notion of security)

e.g. the one-time pad (perfect security),
any AES candidate, DES, etc…

Efficiency

This is the most efficient El Gamal variant:

only 2 exp./Enc and just 1 exp./Dec

- Tsiounis-Yung (PKC '98) D-DH + ROM + Other
  3 exp./Enc - 3 exp./Dec
- Shoup-Gennaro (EC '98) D-DH + ROM
  5 exp./Enc - 7 exp./Dec
- Cramer-Shoup (Crypto '98) D-DH
  5 exp./Enc - 3 exp./Dec
- PSEC-1/2 (PKC '99/Crypto '99) D/C-DH + ROM
  2 exp./Enc - 3 exp./Dec

Com²MaC Workshop on Cryptography

26-28 june 2000 - Pohang - South Korea

Conclusion

David Pointcheval
Département d 'Informatique
ENS - CNRS

David.Pointcheval@ens.fr http://www.di.ens.fr/~pointche

Conclusion

The discrete logarithm setting is very rich:

- One-Way problem ⇒ Secure Signature
- Trapdoor One-Way problem:
  Diffie-Hellman problems
  - computational
  - decisional
  - gap

  ⇒ Secure Encryption
- All are homomorphic

  ⇒ Efficiency